Assessment Report

on

“Predict Disease Outcome Based on Genetic and Clinical Data”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

Name of discipline

By

SHIVAM KUMAR (20240110300233, CSE-AI D)

Under the supervision of

“MR.ABHISHEK SHUKLA”

KIET Group of Institutions, Ghaziabad

May, 2025

Introduction

This project aims to use supervised machine learning to classify patients based on genetic markers, clinical symptoms, and lifestyle factors, predicting whether they are at risk for a particular disease. The dataset includes 30 numerical features that describe characteristics of cell nuclei present in digitized images of breast masses.

Methodology

1. The dataset was cleaned by removing irrelevant columns and handling any missing values.
2. The target column (diagnosis) was encoded using label encoding (M = 1, B = 0).
3. Features and labels were separated, and the data was split into training and testing sets in an 80:20 ratio.

4. A Random Forest Classifier was trained on the dataset using scikit-learn.
5. The model was evaluated using accuracy score and classification report.
6. Feature importance was plotted to visualize the most influential attributes.

Code

```
# Step 1: Install and import required libraries

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.preprocessing import LabelEncoder

from sklearn.metrics import classification_report, accuracy_score


# Step 2: Upload the CSV file

from google.colab import files

uploaded = files.upload()


# Step 3: Load dataset
```

```
file_name = list(uploaded.keys())[0]

df = pd.read_csv(file_name)


# Step 4: Clean dataset

df = df.drop(columns=['id', 'Unnamed: 32'], errors='ignore')  # drop if they exist

df.dropna(inplace=True)  # drop any rows with missing values


# Step 5: Encode target variable

label_encoder = LabelEncoder()

df['diagnosis'] = label_encoder.fit_transform(df['diagnosis'])  # M=1, B=0


# Step 6: Define features and labels

X = df.drop('diagnosis', axis=1)

y = df['diagnosis']


# Step 7: Split data

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


# Step 8: Train Random Forest Classifier

model = RandomForestClassifier(random_state=42)

model.fit(X_train, y_train)

# Step 9: Predict and evaluate

y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)

print(f"Accuracy: {accuracy:.4f}")

print("\nClassification Report:")

print(classification_report(y_test, y_pred))

# Step 10: Plot top 10 feature importances
```

*importances = model.feature_importances_*

feat_imp = pd.DataFrame({'Feature': X.columns, 'Importance': importances}).sort_values(by='Importance', ascending=False)

plt.figure(figsize=(10, 6))

sns.barplot(data=feat_imp.head(10), x='Importance', y='Feature')

plt.title('Top 10 Feature Importances')

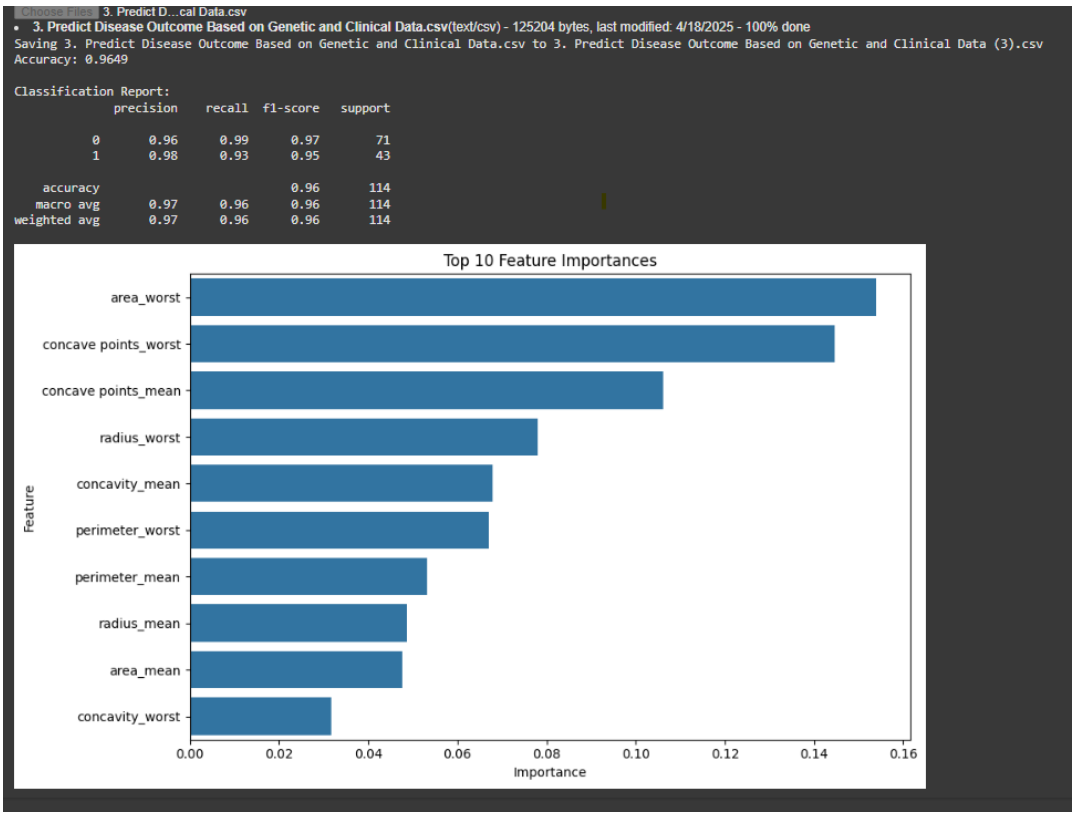
plt.tight_layout()

plt.show()

Output/Result

Accuracy: 96.49%

Classification Report: *(see the screenshot below)*



References / Credits

- Dataset: Breast Cancer Wisconsin (Diagnostic) Dataset
- Libraries: pandas, scikit-learn, matplotlib, seaborn
- Developed and executed using: Google Colab